create database Assignments

--Assignment 1

--1.Date Function Exercises

--Calculate the number of months between your birthday and the current date.

SELECT DATEDIFF(MONTH, '2024-06-10',GETDATE()) AS MonthsBetween;

--Retrieve all orders that were placed in the last 30 days.

SELECT \*

FROM Orders

WHERE OrderDate >= DATEADD(DAY, -30, GETDATE());

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

OrderDate DATE

);

INSERT INTO Orders (OrderID, OrderDate)

VALUES

(1, '2024-03-01'),

(2, '2024-04-15'),

(3, '2024-07-25'),

(4, '2024-07-30'),

(5, '2024-08-20'),

(6, '2024-05-05');

--Write a query to extract the year, month, and day from the current date.

SELECT YEAR(GETDATE()) AS CurrentYear,

MONTH(GETDATE()) AS CurrentMonth,

DAY(GETDATE()) AS CurrentDay;

--Calculate the difference in years between two given dates.

SELECT DATEDIFF(YEAR, '2002-08-10', '2024-06-10') AS YearsDifference;

--Retrieve the last day of the month for a given date.

SELECT EOMONTH('2024-07-21') AS LastDayOfMonth;

--2.String Function Exercises

--Convert all customer names to uppercase.

UPDATE Customers

SET CustomerName = UPPER(CustomerName);

--Extract the first 5 characters of each product name.

select ProductName ,LEFT(ProductName,5) AS ShortName

from Products;

--Concatenate the product name and category with a hyphen in between.

select ProductName ,Category,CONCAT(ProductName,'-',Category) AS ProductDetails

from Products;

--Replace the word 'Phone' with 'Device' in all product names.

select ProductName ,REPLACE(ProductName, 'Phone', 'Device') AS UpdateProductName

from Products;

--Find the position of the letter 'a' in customer names.

SELECT CustomerName,CHARINDEX('a', CustomerName) AS PositionOfA

FROM Customers;

--3.Aggregate Function Exercises

--Calculate the total sales amount for all orders.

select sum(OrderAmt) AS ToatalSalesAmount from Orders;

--Find the average price of products in each category.

select CategoryID, avg(price) AS ProductAvgPrice from Products

GROUP BY CategoryID;

--Count the number of orders placed in each month of the year.

SELECT QrderCount(MONTH, OrderDate) AS Month,

COUNT(\*) AS NumberOfOrders

FROM Orders

ORDER BY Month;

--Find the maximum and minimum order quantities.

SELECT MAX(OrderQuantity) AS MaxOrderQuantity,

MIN(OrderQuantity) AS MinOrderQuantity

FROM Orders;

--Calculate the sum of stock quantities grouped by product category.

select CategotyID, sum(Quantity) AS SumOfStockQuantity

from Products

GROUP BY CategoryID;

--4.Join Exercises

--Write a query to join the Customers and Orders tables to display customer names and their order details.

select c.customername,o.orderid,o.ordername,o.orderdate

from Customers c

inner join Orders o on c.customerid=o.cutomerid;

--Perform an inner join between Products and Orders to retrieve product names and quantities sold.

select p.productname,o.quantity

from Products p

inner join Orders o on p.productid=o.productid;

--Use a left join to display all products, including those that have not been ordered.

select p.productid,p.productname,o.Quantitysold

from Products p

left join Orders o on p.productid=o.productid;

--Write a query to join Employees with Departments and list employee names and their respective department names.

select e.employeename,d.departmentname

from Employees e

join Departments d on e.employeeid =d.employeeid;

--Perform a self-join on an Employees table to show pairs of employees who work in the same department.

SELECT e1.EmployeeName AS Employee1,e2.EmployeeName AS Employee2,e1.DepartmentID

FROM Employees e1

INNER JOIN Employees e2

ON e1.DepartmentID = e2.DepartmentID

AND e1.EmployeeID <> e2.EmployeeID

ORDER BY e1.DepartmentID, e1.EmployeeName, e2.EmployeeName;

--5.Subquery Exercises

--Write a query to find products whose price is higher than the average price of all products.

select productid,productname,price from Products WHERE price >(select avg(price) from Products);

--Retrieve customer names who have placed at least one order by using a subquery.

select customername

from Customers

where customerid IN (select customerid from Orders);

--Find the top 3 most expensive products using a subquery.

select productid,productname,price

from Products

where price in (select top 3 price from Products order by price desc);

order by price desc;

--Write a query to list all employees whose salary is higher than the average salary of their department.

select employeeid, employeename, salary, departmentid

from Employees e1

where salary >(select avg(salary) from Employees e2

where e2.departmentid = e1.departmentid);

--Use a correlated subquery to find employees who earn more than the average salary of all employees in their department.

SELECT EmployeeID,EmployeeName,Salary,DepartmentID

FROM Employees e1

WHERE Salary > (SELECT AVG(Salary) FROM Employees e2

WHERE e2.DepartmentID = e1.DepartmentID);

--6.Grouping and Summarizing Data Exercises

--Group orders by customer and calculate the total amount spent by each customer.

select customerid,sum(amount) as Totalamount

from Orders

group by customerid;

--Group products by category and calculate the average price for each category.

select categoryid,avg(price) as avgprice

from Products

group by categoryid;

--Group orders by month and calculate the total sales for each month.

select orderid,DATEPART(MONTH, OrderDate) AS OrderMonth,,sum(sales) as totalsales

from Orders

group by DATEPART(MONTH, OrderDate);

--Write a query to group products by category and calculate the number of products in each category.

select productid, productname, categoryid, count(productid) as noofproducts

from Products

group by categoryid;

--Use the HAVING clause to filter groups of customers who have placed more than 5 orders.

select customerid ,customername,count(orderid) as ordercount

from Orders

group by customerid

having count(orderid) > 5;

--7.Set Operations (UNION, INTERSECT, EXCEPT)

--Write a query to combine the results of two queries that return the names of customers from different tables using UNION.

select customername from Customers

union

select cutomername from Orders

--Find products that are in both the Electronics and Accessories categories using INTERSECT.

select productname

from Products

where categotyname='Electronics';

intersect

select productname

from Products

where categotyname='Accessories';

--Write a query to find products that are in the Electronics category but not in the Furniture category using EXCEPT.

select productname

from Products

where categotyname='Electronics';

except

select productname

from Products

where categotyname='Furniture';